The Journal of Supercomputing (2023) 79:14981–15003
https://doi.org/10.1007/s11227-023-05250-y

Selection of Six Sigma projects based on integrated multi-criteria decision-making methods: the case of the software development industry

Tülin Erçelebi Ayyıldız[1] · Elmas Burcu Mamak Ekinci[1]

Accepted: 2 April 2023 / Published online: 13 April 2023

Abstract

This study aims to develop a methodology in which alternative Six Sigma projects are prioritized and selected using appropriate multi-criteria decision-making (MCDM) methods in the software development industry. The methodology developed in this paper proposes an MCDM-based approach for researchers to prioritize and select Six Sigma projects for software development projects. The study reveals that by prioritizing software projects with CRITIC, Entropy, and ARAS methods, software companies will be able to achieve their goals such as quality, process improvement, resource allocation, and customer satisfaction. CRiteria Importance Through Intercriteria Correlation (CRITIC) and Entropy methods were used to determine criterion weights and a new Additive Ratio ASsessment (ARAS) method was used to rank alternatives in ordering software development projects. According to the results obtained, one of 7 software development projects (Project 6) was considered the highest priority project for Chidamber and Kemerer's (C&K) software quality metrics. To the best of our knowledge, this is the first study that implements CRITIC and ARAS methods in the Six Sigma project prioritization and selection process for software development projects.

Keywords Project prioritization and selection · Object-oriented software metrics · Software quality · Six sigma · MCDM

✉ Tülin Erçelebi Ayyıldız
ercelebi@baskent.edu.tr

Elmas Burcu Mamak Ekinci
burcuekinci@baskent.edu.tr

[1] Computer Engineering Department, Başkent University, Ankara, Turkey

1 Introduction

Today's enterprises can only survive in a highly competitive environment by providing high-quality products. For improving quality, enterprises should improve their development processes and perform established defect management activities. In this sense quality control methods are important for businesses to achieve sustainable success. There are many popular approaches to quality processes. One of the important aspects of Six Sigma, which is one of them, is that it has an infrastructure that makes measurable improvements within these approaches. The Six Sigma strategy involves the use of statistical tools within a structured methodology to get products and services that are less costly, faster, and better.

Six Sigma is a common approach to the use of statistical tools within a structured methodology to reduce development costs, increase productivity, and develop high-quality products. However, companies have to develop many products at the same time, and conducting Six Sigma projects for all of their products may not be possible due to limited time and resources. Within this context, project prioritization and resource allocation for Six Sigma are two major concerns to be handled with great attention. A systematic Six Sigma project prioritization methodology is the key factor to successfully managing quality improvement activities [18, 54, 55]. There are many studies in the literature stating that Six Sigma project prioritization is the most critical and complicated task of Six Sigma [9, 27, 39, 41, 45, 56]. And also, the success of quality improvement via Six Sigma may relate to the accuracy of utilized project prioritization methods [1, 2, 4, 30–32, 37, 45–47, 50].

The quality of the software is a very important issue for software engineers. As the quality of the software increases, the complexity and number of error/fault rate decrease and maintainability increases. This saves enterprises time and cost.

Software development companies are also among the companies that should hold product quality in the foreground, and Six Sigma is one of the preferred methods used for improving the quality of software besides Object-Oriented Analysis, System Development Life Cycle (SDLC) prototyping, Total Quality Management [14, 28, 67]. However, to the best of our knowledge, there is no Six Sigma project selection study in the literature proposed for software products, although there are studies proposing Six Sigma project selection methodologies for various industries. The goal of this study is to fulfill this gap and propose a Six Sigma prioritization methodology for software development projects. For this purpose, we utilized 7 completed industrial software development projects of a CMMI (Capability Maturity Model Integration) Level 3 certified company in Turkey. CMMI is a process-level improvement training and appraisal program. All projects analyzed are entirely similar software projects developed by the same teams and developed by the same organization. Therefore, software architectures and coding styles for these projects are very similar to each other.

Proposing a prioritization methodology has two steps; the first one is to determine the metrics to be used, and the second one is to select a decision-making method to evaluate the obtained metrics.

To measure the quality of software projects, there are many object-oriented software quality metrics in the literature. The most accepted metric set to measure software quality in the literature is the Chidamber and Kemerer (C&K) metric suite [59]. C&K software quality metrics are used, and the values of these metrics are determined by the static code analysis software tool "Understand 6.0" [61]. C&K metric suite consists of 6 metrics. These metrics are Lack of Cohesion in Methods (LCOM), Coupling between Objects (CBO), Weighted Methods per Class (WMC), Depth of Inheritance Tree (DIT), Number of Children (NOC), and Response for a Class (RFC) [12].

There have been a lot of studies conducted to investigate the value of the C&K metrics. The metrics have often been associated with product results such as quality, fault-proneness, complexity measurement, and maintenance effort [35].

Each C&K metric is a criterion for prioritization, and there are 6 C&K metrics. Hence, applying the Multi-Criteria Decision-Making (MCDM) method is a suitable approach to performing a prioritization. MCDM methods are important tools for choosing the best alternative for multiple criteria in decision-making processes in different fields. In this study, the ARAS method was used for the prioritization and selection of software projects. The ARAS method is also an MCDM method that has been introduced to the literature in recent years [69]. As with other MCDM methods, first of all, the CRITIC and Entropy methods were used to determine criterion weights, and a new ARAS method was used to rank alternatives in prioritizing software development projects. According to the weights obtained from the two methods, the ARAS method was applied and the results were compared. We believe that our proposed methodology will be beneficial to software development companies for their resource allocation management in quality improvement processes.

The next sections are organized as follows. First, Sect. 2 briefly presents a literature survey about a general view of Six Sigma in the software development industry. Next, Sect. 3 demonstrates the proposed method in this study. In Sect. 4, results are discussed. Finally, the conclusion is given in Sect. 5.

2 Literature review

Harry and Schroeder, who pioneered Six Sigma at Motorola in the 1980s, described Six Sigma as "a combination of defects and errors in production and service processes to find the reasons and eliminate the cost and process to reduce cycle time, improve efficiency, meet customer expectations and return of investment (ROI)," proposed an approach that focuses on achieving business improvement [24]. The main purpose of the Six Sigma methodology is to enable businesses to provide services with near zero defects (3.4 defects/million parts), and the basic methodology of Six Sigma is DMAIC (Define, Measure, Analyze, Improve, and Control) [7].

The Six Sigma method has been applied in production processes for more than 30 years. However, the adoption of the Six Sigma methodology in the software development industry has been slower [42]. In software development processes, it is aimed to produce software products consistently with a very high level of customer satisfaction and a minimum number of defects. However, there have been difficulties in achieving this goal for years. Pournaghshband and Watson [42] emphasize the expression "Another way of looking at Six Sigma in a software context would be to achieve a defect-removal efficiency level of about 99.9999%. Since the average defect-removal efficiency level in the United States is only about 85% and less than one project in 1000 has ever topped 98%, it can be seen that actual Six Sigma results are beyond the current state of the art."

In the software development industry, Six Sigma is used at the requirement collection stage and provides useful tools for developing a project concept on problem definition and stakeholder analysis of the software project. There is a strong relation between some software characteristics like deceleration cost, error per product, efficiency, cycle time, and sigma level [65].

The use of Six Sigma in software companies encourages software developers to be more process-centered. Thus, it is aimed to reduce the probability of producing errors per product or service. Companies like Allied Signals, Motorola, and General Electric have saved billions of dollars using Six Sigma [22, 23, 25].

Although Six Sigma is a beneficial strategy, companies have to develop many products at the same time, and conducting Six Sigma projects for all of their products may not be possible due to limited time and resources. Therefore, a prioritization methodology for Six Sigma project selection is crucial for companies that aim to improve quality with smart resource management [41, 53]. In addition, some studies on Six Sigma and software development project selection are given in Table 1, and MCDM methods used in Six Sigma project selection are given in Table 2.

Six Sigma project prioritization and selection methods were reviewed in a recent survey paper comprehensively in [38]. In Pakdil [38], it is stated that decision-makers extensively use MCDM and the researchers recommend using MCDM methods for prioritizing and selecting projects [30, 36, 64, 66]. MCDM methods are used to solve decision problems involving multiple criteria. Thus, it is ensured that the alternatives are ranked by considering the determining criteria. In our study, we also choose MCDM as the selection process. Hence, in this part, we focused on the remarkable research performed for the Six Sigma project selection and MCDM applications which are related to our work.

In the literature survey given in Pakdil [38], the common MCDM methodologies used for selection processes are stated as AHP [3, 34, 66] fuzzy VIKOR [10, 63], fuzzy TOPSIS [3, 68], DEA [21, 34], ANP [63] and DEMATEL [62, 63].

In addition to the studies given above Tuş and Adalı [60], using a new unified decision-making approach based on CRITIC and Weighted Aggregated Sum Product Assessment (WASPAS) methods for the hospital time and attendance software selection problem. On the other hand, Bošković et al. [5] used CRITIC and ARAS methods for the selection of mobile network operators.

Puzovic et al. [44] developed Product Lifecycle Management (PLM) software selection, as an essential part of the PLM concept implementation. This study integrates the Fuzzy Analytic Hierarchy Process (FAHP) and the Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE). Khan and Ansari [29] proposed an intuitionistic fuzzy (IF) improved score function method to deal with MCDM problems under intuitionistic fuzzy sets (IFSs). Recent studies using the ARAS method in combination with CRITIC and Entropy can be seen in

Table 1 Literature review for Six sigma, project selection for software development

Study number	Authors and year	Aim of the study	Application area
1	Bañuelas et al. (2006) [4]	To identify what criteria are considered for selecting Six Sigma projects and how Six Sigma projects are selected in organizations in the UK	UK Organizations
2	Chauhan and Belokar [11]	To provide the steps for the software companies who want to implement the Six Sigma approach for the improvement of their processes in the management of the projects	Software development-Quality control
3	Condé and Martens [13]	To contribute in that direction, aiming to propose a method to carry out the generation and selection of Six Sigma projects	Literature review
4	Shukla et al. [52]	To provide weight and prioritize these criteria for successful project selection	Manufacturing
5	Sarpiri and Gandomani [49]	To attempt was made to present the hybrid software development approach, including Scrum, as the most common agile and Six Sigma methodology	Software development

Table 2 Review of MCDM methods used in Six Sigma project selection

Study number	Authors and year	Aim of the study	Application area	Method
1	Pakdil [38]	Given the critical role of project prioritization and selection process in Six Sigma efforts, this study aims to analyze the relevant literature to answer this question: What types of project prioritization and selection methods have been used in Six Sigma research?	Project prioritization	Literature review
2	Boskovic et al. (2021) [5]	To provide insight into the implementation of Multi-Criteria Decision-Making (MCDM) in the field of MNO evaluation and selection. The paper aims to show how the combination of the CRITIC ARAS methodology can affect the rank of the MNO selection process	Mobile Network Operator	CRITIC, ARAS
3	Sing et al. (2021) [54]	To be carried out to determine the actual project selection process at the industry level	Manufacturing	BWM
4	Tuş and Adalı (2019) [60]	To provide e an approach based on Criteria Importance Through Inter-criteria Correlation (CRITIC) and Weighted Aggregated Sum Product Assessment (WASPAS) methods is used for the time and attendance software selection problem of the private hospital	Health service, software	CRITIC, WASPAS
5	Puzoviç et al. (2019) [44]	To use MCDM models for the multi-aspect assessment of the considered software, and their ranking based upon the overall suitability according to the global goal	Software development	Fuzzy AHP, PROMETHEE
6	Yousefi and Vencheh (2016) [66]	To reduce the risk of failure in Six Sigma projects and select optimal ones which have the most profits and the least expected risks as the first step	Locomotives industry	AHP, TOPSIS, DEA

Table 2 (continued)

Study number	Authors and year	Aim of the study	Application area	Method
7	Rathi et al. [45]	To discuss aimed an approach for the right Six Sigma project selection in the automotive industry using the fuzzy logic-based TOPSIS method	Automotive industry	Fuzzy logic-based TOPSIS, MDL
8	Wang et al. [63]	To apply a Hybrid MCDM Model for Six Sigma Project Selection	The printing industry in Taiwan	DEMATEL, ANP, VİKOR

Table 3. However, to the authors' knowledge, there are no studies in the literature that use CRITIC-ARAS and Entropy-ARAS to rank software development projects.

In the CRITIC method, objective weights are obtained by using the actual data of each evaluation criterion. Therefore, it is sufficient to use the data in the decision matrix to calculate the criterion weights [15]. In this method, correlations between criteria explain how much each criterion affects the other in the decision process. Thus, the weights are obtained objectively for an unbiased ranking that eliminates the evaluation effect. In this way, subjective weights of criteria that are not based on expert opinion can be obtained [15, 60]. Entropy is an objective weighting method like a CRITIC, and it is used to measure the amount of information available from the decision matrix for each criterion [16, 51]. On the other hand, the ARAS method identifies a utility function value for each criterion. This value is directly proportional to the relative effect of the criteria values and weights considered in a project [69]. ARAS method was preferred in this study because it has advantages such as being able to be used when the criteria have different measurement units, being easy to implement, and reflecting the difference between the alternatives and the ideal solution [33].

Our study is the first study that implements and compared the results of the ARAS method with CRITIC and Entropy in the Six Sigma project prioritization and selection process for software development projects.

We analyzed 7 completed industrial software development projects of a CMMI Level 3 certified industry company in Turkey to evaluate our proposed methodology. To select and prioritize these 7 software development projects, we used the C&K metric suite, which is mostly used in the literature. These metrics were measured using a static code analysis tool called Understand, and the results obtained were discussed. The set of C&K metrics used in the study consists of 6 metrics. WMC metric is defined as the number of methods in a class. As the number of methods in a class increases, so does the potential impact on child classes, since child classes inherit all defined methods. An increase in the average WMC increases the error density and decreases the quality of the software [12, 17, 71]. DIT metric is the value of the distance between the highest class and the lowest class of a class in the software. The greater the inherited tree depth for a class, the more difficult it is to predict behavior based on the interaction between inherited features and new features [57]. NOC is the metric describing the number of derived subclasses in software. Having more than one subclass of a class results in high reuse and high error risk. With this criterion, it is possible to measure the features of the software such as efficiency, testability, and reusability [57]. RFC metric specifies the total number of methods in the software class, and these method numbers are potentially seen as invoked by a class. If a class calls many methods, it indicates how high the complexity of the class is [17, 57, 72]. CBO metric is the number of classes on which a class is dependent [72]. As CBO increases, the reusability of a class is likely to decrease. In general, CBO values for each class should be kept low [36]. LCOM metric measures the similarity of methods with each other. The LCOM value should be kept low to keep the agreement high between classes [43].

Thapaliyal et al. [58] stated that a high WMC value increases the error density and decreases the quality. Breesam [6] reported that the depth of the tree indicates

Table 3 Review of recent studies using the ARAS method with CRITIC and entropy

Study number	Authors and year	Aim of the study	Application area	Method
1	Boskovic et al. (2021) [5]	To provide insight into the implementation of Multi-Criteria Decision-Making (MCDM) in the field of MNO evaluation and selection. The paper aims to show how the combination of the CRITIC ARAS methodology can affect the rank of the MNO selection process	Mobile Network Operator	CRITIC, ARAS
2	Jovcic and Prusa (2021) [26]	To decide companies in the logistics industry how can decide on the 3PL evaluation and selection process for outsourcing activities	Logistic	CRITIC, Entropy, ARAS
3	Goswami and Behera [19]	To emphasize the importance of selecting the best suitable material for engineering applications among a group of seven alternatives on the basis of six important criteria, i.e., bending fatigue limit, core hardness, cost, surface hardness, ultimate tensile strength, and surface fatigue limit by analyzing through multi-criteria decision-making (MCDM) techniques	Mechanical and Material Engineering	Entropy, ARAS
4	Goswami et al. [20]	To create two novel hybrid MCDM systems by integrating Additive Ratio ASsessment (ARAS) with Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and Complex PRoportional ASsessment (COPRAS)	Mechanical Engineering	ARAS, COPRAS, TOPSIS, CRITIC
5	Ecer [16]	The aim of this study is to analyze the corporate sustainability performance of privately owned banks in Turkey	Finance, Sustainability	Entropy, ARAS

the complexity of the design, that it is difficult to understand the system with many inheritance layers, and that the high DIT criterion increases errors. Chidamber and Kemerer [12] stated that a high number of subclasses, that is, the NOC criterion, generally means that it is more complex, more difficult to maintain, and has a higher tendency to error its value should below. At the same time, they also stated that triggering too many methods in a message sent for RFC criteria makes it difficult to detect the errors in the classes and to test the classes. They stated that low compatibility for the LCOM criterion increases the complexity, and therefore, errors in the development phase will increase.

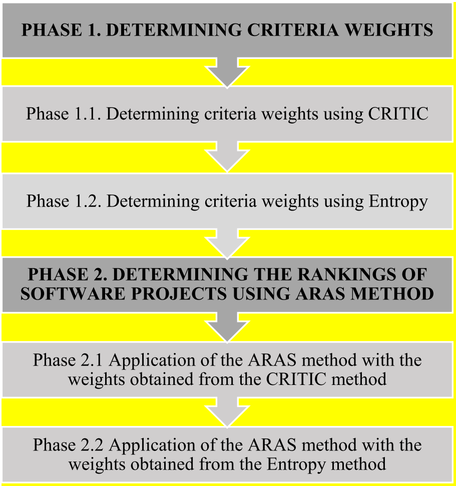
Apart from these mentioned studies, Briand et al. [8] showed that CBO, RFC, and LCOM criteria can be used to find the error tendency of classes by using the logistic regression method. In the study by Zhou et al. [72], it was shown that C&K criteria other than DIT criteria and the number of lines of code are important in determining error trends.

The papers mentioned above are mainly manufacturing-oriented and none of them include the Six Sigma project selection method in the software development industry, as a gap in the literature. To fulfill this gap, we propose a Six Sigma prioritization methodology for software development projects. For this purpose, we utilized 7 completed industrial software development projects of a CMMI Level 3 certified company in Turkey.

3 Research methodology

The prioritization and selection of software projects were carried out in two phases. First, the criteria weights were determined using the CRITIC and Entropy methods, and then, the software projects were ranked according to the C&K metrics using the ARAS method for the weights obtained from both methods (Fig. 1).

Fig. 1 Steps of analysis


PHASE 1. DETERMINING CRITERIA WEIGHTS
Phase 1.1. Determining criteria weights using CRITIC
Phase 1.2. Determining criteria weights using Entropy
PHASE 2. DETERMINING THE RANKINGS OF SOFTWARE PROJECTS USING ARAS METHOD
Phase 2.1 Application of the ARAS method with the weights obtained from the CRITIC method
Phase 2.2 Application of the ARAS method with the weights obtained from the Entropy method

3.1 Phase 1.1 Determining criteria weights using CRITIC

The CRITIC method was developed by Diakoulaki et al. [15]. In this study, instead of subjective weights obtained from expert opinion, the CRITIC method, in which objective weights are obtained using the correlation between standard deviation and criteria, was preferred [15]. The application steps of the CRITIC method are given below [15, 40, 48]. CRITIC method was applied to compute the importance weights of C&K metrics.

3.1.1 Step 1.1.1 Determining the initial decision matrix

Let $[X]$ denotes the initial decision matrix, it can be represented as in Eq. (1)

$$[X] = \left[x_{ij}\right]_{n\times m} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{bmatrix} \tag{1}$$

where x_{ij} is the element of $[X]$, i is the number of alternatives $(A_i, i = 1, 2, \dots, n)$, and j is the number of criteria $\left(C_j, j = 1, 2, \dots, m\right)$.

In the application, data are obtained for seven software projects $(A_i, i = 1, 2, \dots, 7)$ for six criteria $\left(C_j, j = 1, 2, \dots, 6\right)$ named as C&K metrics in Sect. 1. The considered criteria are as follows: WMC (C_1), DIT (C_2), NOC (C_3), CBO (C_4), RFC (C_5), LCOM (C_6). All criteria are cost-type criteria. The initial decision matrix in Table 4 has been created with the mean of the metric values obtained for all classes of software projects by the "Understand 6.0" static software code analysis tool.

3.1.2 Step 1.1.2. Normalizing the initial decision matrix

The initial decision matrix is normalized for benefit and cost type criteria, respectively, Eqs. (2) and (3).

Table 4 Initial decision matrix

Software projects	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
A_1	11.071	2.522	0.732	11.047	34.362	89.148
A_2	58.384	2.727	0.712	12.217	76.875	36.727
A_3	11.284	1.793	0.369	13.233	34.037	131.580
A_4	12.352	2.368	0.447	14.317	39.850	259.907
A_5	13.509	1.869	0.738	10.120	30.351	100.464
A_6	12.959	1.688	0.364	7.573	33.471	176.276
A_7	9.022	1.677	0.437	10.808	22.978	80.341

$$n_{ij} = \frac{x_{ij} - \min\left(x_j\right)}{\max\left(x_j\right) - \min\left(x_j\right)} \tag{2}$$

$$n_{ij} = \frac{\max\left(x_j\right) - x_{ij}}{\max\left(x_j\right) - \min\left(x_j\right)} \tag{3}$$

where n_{ij} is normalized values of x_{ij}, min(x_j) is the minimum value of jth criterion by alternatives, and max(x_j) is the maximum value of jth criterion by alternatives.

The normalized initial decision matrix in Table 5 is obtained by Eq. (3) since all criteria are cost types.

3.1.3 Step 1.1.3. Determining the level of relationship between the criteria

Multiple correlation coefficients are calculated using the normalized decision matrix to determine the level of correlation between the criteria, and a $m \times m$ type correlation matrix is obtained. The Pearson correlation coefficient gives the linear relationship between the two variables under the assumption that the variables have a normal distribution [7]. In this study, since the values of criteria for the alternatives are not normally distributed, the Spearman correlation coefficient based on elements of rank orders given in Eq. (4) was used to calculate the relationship criteria. The correlation matrix for criteria is given in Table 6.

$$\rho_{jk} = 1 - \frac{\sum_{i=1}^{n} d_i^2}{n\left(n^2 - 1\right)}, \quad (j, k = 1, 2, \ldots, m) \tag{4}$$

3.1.4 Step 1.1.4. Calculate the amount of information (S_j)

Standard deviation is used to measure contrast intensity and obtain objective criteria weights [70]. Thus, it can be said that the scores of the alternatives with higher weights differ from the weights obtained. Similarly, it can be argued that a criterion in which all alternatives have the same performance does not provide

Table 5 Normalized initial decision matrix

Software projects	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
A_1	0.958	0.195	0.016	0.485	0.789	0.969
A_2	0.000	0.000	0.068	0.311	0.000	0.000
A_3	0.954	0.890	0.985	0.161	0.795	0.820
A_4	0.933	0.342	0.775	0.000	0.687	0.369
A_5	0.909	0.817	0.000	0.622	0.863	0.929
A_6	0.920	0.990	1.000	1.000	0.805	0.663
A_7	1.000	1.000	0.805	0.520	1.000	1.000

Table 6 Relationship coefficient matrix

Criteria	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
WMC	1.000	0.464	0.250	−0.107	0.392	0.750
DIT	0.464	1.000	0.643	0.464	0.857	0.500
NOC	0.250	0.642	1.000	0.036	0.179	−0.143
CBO	−0.107	0.464	0.036	1.000	0.679	0.393
RFC	0.392	0.857	0.179	0.679	1.000	0.750
LCOM	0.750	0.500	−0.143	0.393	0.750	1.000

any additional information and is of no use to be included in the decision-making process [15].

The total information amount in the criteria is given in Eq. (5), and the measure of contrast intensity is given with the standard deviation calculated by Eq. (6).

$$S_j = \sigma_j \sum_{k=1}^{m} \left(1 - \rho_{jk}\right), \quad j = 1, 2, \ldots, m \tag{5}$$

$$\sigma_j = \sqrt{\frac{\sum_{i=1}^{n} \left(n_{ij} - \overline{n}_j\right)^2}{n}}, \quad j = 1, 2, \ldots, m \tag{6}$$

The standard deviation and total information are given in Table 7.

3.1.5 Step 1.1.5 Calculate the criteria weight (w_j)

Criteria weight for each criterion is computed by Eq. (7), and w_j for each metric is obtained as shown in Table 8.

Table 7 Values of $(1 - \rho_{jk})$, S_j and σ_j

Criteria	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
WMC	0.000	0.536	0.75	1.107	0.608	0.250
DIT	0.536	0.000	0.357	0.536	0.143	0.500
NOC	0.750	0.358	0.000	0.964	0.821	1.143
CBO	1.107	0.536	0.964	0.000	0.321	0.607
RFC	0.608	0.143	0.821	0.321	0.000	0.250
LCOM	0.250	0.500	1.143	0.607	0.250	0.000
S_j	1.166	0.860	1.894	1.158	0.697	1.021
σ_j	0.359	0.415	0.469	0.327	0.325	0.371

Table 8 Criteria weights

Criteria	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
w_j	0.172	0.127	0.279	0.170	0.103	0.150

$$w_j = \frac{S_j}{\sum_{k=1}^{m} S_k}, \quad j = 1, 2, \ldots, m \tag{7}$$

3.2 Phase 1.2 Determining criteria weights using entropy

3.2.1 Step 1.2.1 Determining initial decision matrix

Let $[X]$ denotes the initial decision matrix, it can be represented as in Eq. (1). The initial decision matrix in Table 4 is used.

3.2.2 Step 1.2.2 Normalizing the initial decision matrix

Elements of normalized initial decision matrix $[Z]$ denoted as $z_{ij}; i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots ., m$ shows the normalized value of the *jth* criterion for the *ith* software.

$$z_{ij} = \frac{x_{ij}}{\sum_{i=1}^{n} x_{ij}} \tag{8}$$

The normalized initial decision matrix in Table 9 is obtained by Eq. (8)

3.2.3 Step 1.2.3 Calculated entropy values and importance weight

Entropy value for each criterion $e_j; j = 1, 2, \ldots, m$ was calculated with Eq. (9) and is given in Table 10.

Table 9 Normalized initial decision matrix

Software projects	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
A_1	0.086	0.172	0.193	0.139	0.126	0.074
A_2	0.454	0.186	0.188	0.154	0.283	0.303
A_3	0.088	0.122	0.097	0.167	0.125	0.109
A_4	0.096	0.162	0.118	0.181	0.147	0.216
A_5	0.105	0.128	0.194	0.128	0.112	0.084
A_6	0.101	0.115	0.096	0.095	0.123	0.147
A_7	0.070	0.115	0.115	0.136	0.085	0.067

Table 10 Entropy value and importance weight for each criterion

C_1	C_2	C_3	C_4	C_5	C_6
$\boldsymbol{e}_1$	$\boldsymbol{e}_2$	$\boldsymbol{e}_3$	$\boldsymbol{e}_4$	$\boldsymbol{e}_5$	$\boldsymbol{e}_6$
0.413	0.269	0.277	0.269	0.290	0.312
$\boldsymbol{w}_1$	$\boldsymbol{w}_2$	$\boldsymbol{w}_3$	$\boldsymbol{w}_4$	$\boldsymbol{w}_5$	$\boldsymbol{w}_6$
0.141	0.175	0.174	0.175	0.170	0.165

$$e_j = \frac{-\sum_{i=1}^{n}\left[z_{ij}\ln\left(z_{ij}\right)\right]}{\ln(n)}. \tag{9}$$

For each criterion, importance weight $w_{j;}\ j = 1, 2, \ldots ., m$ was calculated by Eq. (10) and is given in Table 10.

$$w_j = \frac{1 - e_j}{\sum_{j=1}^{n} 1 - e_j}. \tag{10}$$

3.3 Phase 2: Determining the rankings of software projects using the ARAS method

The ARAS method was presented in the literature by Zavadskas and Turskis [69]. This method compares the utility function values of the decision alternatives in the decision problem with the utility function value of the optimal alternative added to the decision problem by the decision maker Zavadskas and Turskis [69].

3.3.1 Step 2.1 Formulating the initial decision matrix

The initial decision matrix is given in Table 4 using Eq. (1). In the ARAS method, unlike other MCDM methods, there is a row consisting of the optimal values of each criterion in the initial decision matrix. If the optimal value of the j criterion is unknown, then

$$x_{0j} = \begin{cases} \max\limits_i x_{ij}, & \text{if } \max\limits_i x_{ij}\ is \text{ preferable}, \\ \min\limits_i x_{ij}, & \text{if } \min\limits_i x_{ij}\ is \text{ preferable}, \end{cases} \tag{11}$$

As shown in Table 11, since the optimal values are not known for the criteria used to measure the software quality, and all criteria are cost types, the minimum value of alternatives for each criterion was taken as the optimal value.

Table 11 Optimal values

Software projects	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
Optimal value (x_{0j})	9.022	1.677	0.364	7.573	22.978	80.341

3.3.2 Step 2.2. Normalize the initial decision matrix

The initial decision matrix in Table 4 is normalized as in Eq. (12), and the normalized initial decision matrix in Table 12 is obtained.

$$\bar{x}_{ij} = \begin{cases} \frac{x_{ij}}{\sum_{i=0}^{n} x_{ij}}, & \text{benefit type criteria} \\ \frac{y_{ij}}{\sum_{i=0}^{n} y_{ij}}, & \text{cost type criteria} \end{cases} \tag{12}$$

where $y_{ij} = \frac{1}{x_{ij}}$.

3.3.3 Step 2.3 Calculating weighted normalized initial decision matrix

Let $\left[\hat{X}\right]$ denotes the normalize-weighted matrix as in Eq. (13)

$$\left[\hat{X}\right] = \left[\hat{x}_{ij}\right]_{n\times m} = \begin{bmatrix} \hat{x}_{11} & \hat{x}_{12} & \cdots & \hat{x}_{1m} \\ \hat{x}_{21} & \hat{x}_{22} & \cdots & \hat{x}_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ \hat{x}_{n1} & \hat{x}_{n2} & \cdots & \hat{x}_{nm} \end{bmatrix} \tag{13}$$

where weighted normalized values of all the criteria are calculated by

$$\hat{x}_{ij} = \bar{x}_{ij} w_j, \quad i = 0, 1, \dots, n \tag{14}$$

and w_j is the criteria importance weights obtained from CRITIC in Phase 1.1 and Entropy in Phase 1.2. and weighted normalized initial decision matrix is in Tables 13 and 14, respectively.

3.3.4 Step 2.4 Determining optimality function values

Optimality function values are calculated as follows

Table 12 The normalized initial decision matrix

Software projects	C_1	C_2	C_3	C_4	C_5	C_6
	WMC	DIT	NOC	CBO	RFC	LCOM
A_1	0.139	0.098	0.081	0.117	0.118	0.170
A_2	0.026	0.090	0.083	0.106	0.053	0.042
A_3	0.136	0.137	0.161	0.098	0.119	0.115
A_4	0.125	0.104	0.133	0.090	0.102	0.058
A_5	0.114	0.132	0.080	0.128	0.134	0.151
A_6	0.119	0.146	0.163	0.171	0.121	0.086
A_7	0.171	0.147	0.136	0.120	0.177	0.189
Optimal value (x_{0j})	0.171	0.147	0.163	0.171	0.177	0.189

Table 13 Weighted normalized initial decision matrix with CRITIC

Software projects	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
A_1	0.024	0.012	0.023	0.020	0.012	0.026
A_2	0.005	0.011	0.023	0.018	0.005	0.006
A_3	0.023	0.017	0.045	0.017	0.012	0.017
A_4	0.021	0.013	0.037	0.015	0.010	0.009
A_5	0.020	0.017	0.022	0.022	0.014	0.023
A_6	0.020	0.018	0.045	0.029	0.012	0.013
A_7	0.029	0.019	0.038	0.020	0.018	0.028
Optimal value (x_{0j})	0.029	0.019	0.045	0.029	0.018	0.028

Table 14 Weighted normalized initial decision matrix with entropy

Software projects	C_1 WMC	C_2 DIT	C_3 NOC	C_4 CBO	C_5 RFC	C_6 LCOM
A_1	0.020	0.017	0.014	0.021	0.020	0.028
A_2	0.004	0.016	0.014	0.019	0.009	0.007
A_3	0.019	0.024	0.028	0.017	0.020	0.019
A_4	0.018	0.018	0.023	0.016	0.017	0.010
A_5	0.016	0.023	0.014	0.022	0.023	0.025
A_6	0.017	0.026	0.028	0.030	0.021	0.014
A_7	0.024	0.026	0.024	0.021	0.030	0.031
Optimal value (x_{0j})	0.024	0.026	0.028	0.030	0.030	0.031

Table 15 Optimality function values

Software projects	O_i for CRITIC	O_i for Entropy
A_1	0.116	0.119
A_2	0.069	0.068
A_3	0.131	0.128
A_4	0.106	0.102
A_5	0.117	0.123
A_6	0.139	0.135
A_7	0.153	0.156
Optimal value (x_{0j})	0.169	0.169

$$O_i = \sum_{j=1}^{m} \hat{x}_{ij}, \quad i = 0, 1, 2, \dots, n \tag{15}$$

where O_i is the value of the optimality function of ith alternative for each weighting method. Optimality function values are given in Table 15.

3.3.5 Step 2.5 Calculating utility degree

Alternative utility degree is explained by a comparison of the variant, which is analyzed, with the ideally best one O_0. Utility degree calculated by Eq. (16) for each alternative.

$$K_i = \frac{O_i}{O_0}, \quad i = 1, \ldots, n \tag{16}$$

where O_i and O_0 are the optimality function values from Eq. (15)

To measure and interpret the quality of software development projects with C&K metrics, a threshold value has to be determined for each metric. However, it is possible to encounter very different threshold values in the literature. Therefore, we used criteria weights which are given in Tables 8 and 10. According to Table 8 which was calculated by CRITIC, the NOC metric's weight is the highest. However, it is seen that the criteria weights calculated by the entropy method are close to each other. The utility degree and ranks of the projects are given in Table 16.

According to the results obtained, as shown in Table 10, out of 7 software development projects, Project 7 was considered the highest priority project for two weighting methods. When the C&K metric values of Project 7 were examined, it was seen that after applying the MCDM method optimal value was highest compared to other projects.

4 Results and discussion

To ensure continuous development in areas where technology is developed, variables must be measurable. In this context, the measurability of software metrics in improving software quality enables the use of quantitative decision-making methods and objective evaluations. This leads to the selection of the right project within the scope of continuous improvement, thus preventing cost and time loss.

It is very crucial to prioritize the projects for the Six Sigma application to save time and allocate resources efficiently. In this study, we proposed a prioritization

Table 16 Utility degree in CRITIC and entropy

Weighting method	CRITIC		Entropy	
Software projects	K_i	Ranks	K_i	Ranks
A_1	0.689	5	0.706	5
A_2	0.408	7	0.404	7
A_3	0.780	3	0.754	3
A_4	0.628	6	0.600	6
A_5	0.692	4	0.728	4
A_6	0.822	2	0.800	2
A_7	0.903	1	0.919	1
Optimal value (x_{0j})	1.000		1.0000	

method for software development projects. Prioritization was performed based on the 6 C&K metrics. Since there is no precedence of any of the C&K metrics over any other, it is not possible to prioritize the software projects by just looking at the C&K metrics measurements. The MCDM-based prioritizing methodology allows the altogether evaluation of these 6 C&K metrics analytically. As shown in Table 16, Project 7 was found to have the highest priority for Six Sigma application among 7 software development projects after the application of our proposed methodology for two weighting methods.

The application of the methodology was shown step by step so that any practitioner can apply the proposed process easily. Our proposed method can be applied to all object-oriented software projects since the C&K metrics can easily be obtained by using static code analysis tools such as Understand and SonarQube. Since these metrics are obtained without user interpretation, the prioritization results are also objective, making the proposed methodology reliable as well.

We used CRITIC, Entropy, and ARAS as the MCDM methods. Other MCDM methods may also be used for selection. We selected CRITIC, Entropy, and ARAS due to their objective selection opportunities. These methods also provide the objective ranking based on initial decision matrix values.

Our proposed methodology fulfills the gap of not having a Six Sigma project selection methodology in the software development industry. We believe that our proposed methodology will help the decision-makers in decreasing their effort and wasting resources on quality improvement activities.

5 Conclusion

Six Sigma is a common approach used for quality improvement. Unfortunately, companies may not apply Six Sigma to all of their projects due to limited time and resources. So, the companies are obliqued to make a selection among the projects to which they can apply Six Sigma. A methodological way of prioritizing projects for Six Sigma selection is crucial for effective resource management.

In this study, we aimed to propose a prioritization methodology for Six Sigma selection in software development projects. For this purpose, we utilized 7 completed industrial software development projects of a CMMI Level 3 certified company in Turkey. The methodology used in our proposal is based on CRITIC, Entropy, and ARAS, which are MCDM-based methods. To the best of our knowledge, our study is the first one which proposes a prioritization methodology for Six Sigma selection in software development projects. Besides, our study is the first study that implements CRITIC, Entropy, and ARAS MCDM methods in selecting software development projects. We used C&K metrics to measure the quality of software projects. C&K metrics provide an interpretation of software quality characteristics such as efficiency, complexity, understandability, reusability, and maintainability. The values of C&K metrics are measured by the static code analysis tool, Understand.

Since the projects selected in the study are from the same field, it is thought that the project selected for the implementation of Six Sigma as a result of the prioritization will also be a pioneer in the improvement studies of the other projects.

Our study is explained in a systematic way giving the implementation steps. We believe that these steps will help software companies achieve their goals such as quality, process improvement, resource allocation, and customer satisfaction efficiently.

In future work, we are planning to increase the number of projects and apply different MCDM methods to the different industrial sectors.

Authors' contributions Both authors equally contributed their skills and effort to produce this article.

Funding Not applicable.

Availability of data and materials Data are available from the authors upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval Not applicable.

References

1. Adams C, Gupta P, Wilson C (2003) Six sigma deployment. Butterworth-Heinemann, Oxford
2. Bakar FAA, Subari K, Daril MAM (2015) Critical success factors of lean six sigma deployment: a current review. Int J Lean Six Sigma 6(4):339–348
3. Bañuelas R, Antony J (2003) Going from six sigma to design for six sigma: an exploratory study using analytic hierarchy process. TQM Mag 15(5):334–344
4. Bañuelas R, Tennant C, Tuersley I, Tang S (2006) Selection of six sigma projects in the UK. TQM Mag 18(5):514–527
5. Bošković S, Radonjić-Djogatović V, Ralević P, Dobrodolac M, Jovčić S (2021) Selection of mobile network operator using the critic-aras method. Int J Traffic Transp Eng 11:17–29
6. Breesam KM (2007) Metrics for object-oriented design focusing on class inheritance metrics. In: 2nd International Conference on Dependability of Computer Systems, 14–16 June 2007, Szklarska, Poland
7. Breyfogle FW III (1999) Implementing Six Sigma: smarter solutions using statistical methods. John Wiley and Sons, Hoboken
8. Briand LC, Wust J, Ikonomovski SV, Lounis H (1999) Investigating quality factors in object-oriented designs: an industrial case study. In: Proceedings of the 1999 International Conference on Software Engineering, 22 May 1999, Los Angeles, CA, USA
9. Büyüközkan G, Öztürkcan D (2010) An integrated analytic approach for six sigma project selection. Expert Syst Appl 37(8):5835–5847
10. Çakir E (2017) Application of fuzzy multi-criteria decision making methods on six sigma projects selection. J Soc Adm Sci 4(1):132–138
11. Chauhan Y, Belokar RM (2015) Six Sigma in Project Management for Software Companies. Available at SSRN: https://ssrn.com/abstract=2624188 or https://doi.org/10.2139/ssrn.2624188
12. Chidamber SR, Kemerer CF (1994) A metrics suite for object-oriented design. IEEE Trans Softw Eng 20(6):476–493. https://doi.org/10.1109/32.295895
13. Condé GCP, Martens ML (2020) Six sigma project generation and selection: literature review and feature based method proposition. Prod Plan Control 31(16):1303–1312
14. Dennis A, Wixom B, Tegarden D (2004) Systems analysis and design: an object oriented approach with UML version 2.0. Wiley, New York

15. Diakoulaki D, Mavrotas G, Papayannakis L (1995) Determining objective weights in multiple criteria problems: the critic method. Comput Oper Res 22(7):763–770. https://doi.org/10.1016/0305-0548(94)00059-H
16. Ecer F (2019) A multi-criteria approach towards assessing corporate sustainability performances of privately-owned banks: entropy-ARAS integrated model. Eskiseh Osman Univ IIBF Derg-Eskiseh Osman Univ J Econ Adm Sci 14:365–390
17. Fenton NE, Martin N (1999) Software metrics: successes, failures and new directions. J Syst Softw 47(2–3):149–157. https://doi.org/10.1016/S0164-1212(99)00035-7
18. Gijo EV, Rao TS (2005) Six sigma implementation—hurdles and more hurdles. Total Qual Manag Bus Excell 16(6):721–725
19. Goswami SS, Behera DK (2021) Implementation of ENTROPY-ARAS decision making methodology in the selection of best engineering materials. Mater Today Proc 38:2256–2262
20. Goswami SS, Behera DK, Afzal A, Razak Kaladgi A, Khan SA, Rajendran P, Subbiah R, Asif M (2021) Analysis of a robot selection problem using two newly developed hybrid MCDM models of TOPSIS-ARAS and COPRAS-ARAS. Symmetry 13(8):1331
21. Hadi-Vencheh A, Yousefi A (2018) Selecting six sigma project: a comparative study of DEA and LDA techniques. Int J Lean Six Sigma 9(4):506–522
22. Hahn GJ et al (1999) The impact of six sigma improvement—a glimpse into the future of statistics. Am Stat 53:1–8
23. Hahn GJ et al (2000) The evolution of six sigma. Qual Eng 12(3):317–326
24. Harry MJ, ve Schroeder RR (2000) Six Sigma: the breakthrough management strategy revolutionizing the world’s top corporations. Dell Publishing Group Inc, New York
25. Hoerl RW (1998) Six sigma and the future of the quality profession. Qual Prog 31(6):35–42
26. Jovčić S, Průša P (2021) A hybrid MCDM approach in third-party logistics (3PL) provider selection. Mathematics 9(21):2729
27. Kalashnikov V, Benita F, Lopez-Ramos F, Hernandez-Luna A (2017) Bi-objective project portfolio selection in lean six sigma. Int J Prod Econ 186:81–88
28. Kendall D, Kendall J (2004) Systems analysis and design. Prentice Hall, Clifton Woods
29. Khan M, Ansari MD (2020) Multi-criteria software quality model selection based on divergence measure and score function. J Intell Fuzzy Syst 38(3):3179–3188. https://doi.org/10.3233/JIFS-191153
30. Kornfeld B, Kara S (2013) Selection of lean and six sigma projects in industry. Int J Lean Six Sigma 4(1):1071–1088
31. Kumar M, Antony J, Cho BR (2009) Project selection and its impact on the successful deployment of six sigma. Bus Process Manag J 15(5):669–686
32. Kwak YH, Anbari FT (2006) Benefits, obstacles, and future of six sigma approach. Technovation 26(5/6):708–715
33. Liu N, Xu Z (2021) An overview of ARAS method: theory development, application extension, and future challenge. Int J Intell Syst 36(7):3524–3565
34. Mawby WD (2007) Project portfolio selection for Six Sigma. Quality Press
35. Michura J, Capretz MAM, Wang S (2013) Extension of object-oriented metrics suite for software maintenance. Int Sch Res Not. https://doi.org/10.1155/2013/276105
36. Moaven S, Habibi J (2020) A fuzzy-AHP-based approach to select software architecture based on quality attributes (FASSA). Knowl Inf Syst 62:4569–4597. https://doi.org/10.1007/s10115-020-01496-7
37. Padhy R (2017) Six sigma project selections: a critical review. Int J Lean Six Sigma 8(2):244–258
38. Pakdil F (2022) Six sigma project prioritization and selection methods: a systematic literature review. Int J Lean Six Sigma 13(2):382–407. https://doi.org/10.1108/IJLSS-01-2021-0001
39. Pakdil F, Toktaş P, Can GF (2020) Six sigma project prioritization and selection: a multi-criteria decision making approach in the healthcare industry. Int J Lean Six Sigma 12(3):553–578. https://doi.org/10.1108/IJLSS-04-2020-0054
40. Pan B, Liu S, Xie Z, Shao Y, Li X, Ge R (2021) Evaluating operational features of three unconventional intersections under heavy traffic based on CRITIC method. Sustainability 13:4098. https://doi.org/10.3390/su13084098
41. Pande P, Neuman R, Cavanagh R (2000) The Six Sigma way: how GE, Motorola and other top companies are honing their performance. McGraw-Hill, New York

42. Pournaghshband H, Watson J (2017) Should Six Sigma be incorporated into software development & project management? In: International Conference on Computational Science and Computational Intelligence. https://doi.org/10.1109/CSCI.2017.176
43. Pressman RS (2000) Software engineering: a practitioner's approach, 5th edn. McGraw-Hill, New York
44. Puzovic S, Vasovic JV, Radojicic M, Paunovic V (2019) An integrated MCDM approach to PLM software selection. Acta Polytechnica Hungarica 16(4):45–65
45. Rathi R, Khanduja D, Sharma S (2015) Six sigma project selection using fuzzy TOPSIS decision-making approach. Manag Sci Lett 5(5):447–456
46. Rathi R, Khanduja D, Sharma S (2016) A fuzzy MADM approach for project selection: a six sigma case study. Decis Sci Lett 5(2):255–268
47. Saghaei A, Didehkhani H (2011) Developing an integrated model for the evaluation and selection of six sigma projects based on ANFIS and fuzzy goal programming. Expert Syst Appl 38(1):721–728
48. Şahin M (2021) Location selection by multi-criteria decision-making methods based on objective and subjective weightings. Knowl Inf Syst. https://doi.org/10.1007/s10115-021-01588-y
49. Sarpiri MN, Gandomani TJ (2021) A case study of using the hybrid model of scrum and six sigma in software development. Int J Electr Comput Eng (2088–8708) 11(6)
50. Shanmugaraja M, Nataraj M, Gunasekaran N (2012) Six sigma project selection via quality function deployment. Int J Product Qual Manag 10(1):85–111
51. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27(3):379–423
52. Shukla V, Swarnakar V, Singh AR (2021) Prioritization of lean six sigma project selection criteria using best worst method. Mater Today Proc 47:5749–5754
53. Singh M, Rathi R (2019) A structured review of lean six sigma in various industrial sectors. Int J Lean Six Sigma 10(2):622–664
54. Singh K, Swarnakar V, Singh AR (2021) Lean six sigma project selection using best worst method. Mater Today Proc 47:5766–5770
55. Snee R, Rodebaugh WF Jr (2002) The project selection process. Qual Prog 35(9):78–80
56. Su CT, Chou CJ (2008) A systematic methodology for the creation of six sigma projects: a case study of semiconductor foundry. Expert Syst Appl 34(4):2693–2703
57. Subramanyam R, Krishnan MS (2003) Empirical analysis of CK metrics for object-oriented design complexity: implications for software defects. IEEE Trans Softw Eng 29(4):297–310. https://doi.org/10.1109/TSE.2003.1191795
58. Thapaliyal MP, Verma G (2010) Software defects and object-oriented metrics—an empirical analysis. Int J Comput Appl. https://doi.org/10.5120/1379-1859
59. Tiwari S, Rathore SS (2018) Coupling and cohesion metrics for object-oriented software: a systematic mapping study. In: Proceedings of the 11th Innovations in Software Engineering Conference, Hyderabad, India, 9–11 February 2018
60. Tuş A, Aytaç Adalı E (2019) The new combination with CRITIC and WASPAS methods for the time and attendance software selection problem. Opsearch 56(2):528–538
61. Understand User Guide and Reference Manual (2021) Available at: https://documentation.scitools.com/pdf/understand.pdf. Accessed 29 Nov 2021
62. Vinodh S, Swarnakar V (2015) Lean six sigma project selection using hybrid approach based on fuzzy DEMATEL–ANP–TOPSIS. Int J Lean Six Sigma 6(4):313–338
63. Wang FK, Hsu CH, Tzeng GH (2014) Applying a hybrid MCDM model for six sigma project selection. Math Probl Eng 2014:1–13
64. Wei CC, Cheng YL (2020) Six sigma project selection using fuzzy multiple attribute decision-making method. Total Qual Manag Bus Excell 31(11/12):1266–1289
65. Wen Y, An Q, Xu X, Chen Y (2018) Selection of six sigma project with interval data: common weight DEA model. Kybernetes 47(7):1307–1324
66. Yousefi A, Hadi-Vencheh A (2016) Selecting six sigma projects: MCDM or DEA? J Model Manag 11(1):309–325
67. Zahedi F (1995) Quality information systems. Danvers, MA, Boyd and Praser
68. Zardari NH, Ahmed K, Shirazi SM, Yusop ZB (2015) Weighting methods and their effects on multi-criteria decision making model outcomes in water resources management. Springer, Cham
69. Zavadskas EK, Turskis Z (2010) A new additive ratio assessment (ARAS) method in multicriteria decision-making. Technol Econ Dev Econ 16(2):159–172
70. Zeleny M (1982) Multiple criteria decision making. McGraw-Hill, New York

71. Zhao L, Hayes JH (2006) Predicting classes in need of refactoring: an application of static metrics. In: 2nd International PROMISE Workshop, Philadelphia, Pennsylvania USA

72. Zhou Y, Leung H (2006) Empirical analysis of object-oriented design metrics for predicting high and low severity faults. IEEE Trans Softw Eng 32(10):771–789. https://doi.org/10.1109/TSE.2006.102

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.